(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 July 2001 (19.07.2001)

PCT

(10) International Publication Number
WO 01/52568 A2

(51) International Patent Classification[7]: H04Q 7/00

(21) International Application Number: PCT/US01/00896

(22) International Filing Date: 10 January 2001 (10.01.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/175,463 10 January 2000 (10.01.2000) US
09/757,773 9 January 2001 (09.01.2001) US

(71) Applicant: QUALCOMM INCORPORATED [US/US]; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(72) Inventors: FISCHEL, Scott, Eduard; 548 Ford Avenue, Solana Beach, CA 92075 (US). MIR, Idreas, A.; 7532 Charmant Drive #331, San Diego, CA 92122 (US).

(74) Agents: OGROD, Gregory, D. et al.; Qualcomm Incorporated, 5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR TESTING WIRELESS COMMUNICATION CHANNELS

(57) Abstract: Techniques to test a wireless communication link. A traffic channel is tested via a test data service option (TDSO) that may be negotiated and connected similar to other services. Test parameters values may be proposed, accepted or rejected, and negotiated. Test data for a channel is generated based on a defined data pattern or a pseudo-random number generator. Sufficient test data may be generated based on the generator for a test interval, stored to a buffer, and thereafter retrieved from a particular section of the buffer to form data block(s) for each "active" frame. The traffic channel may be tested using discontinuous transmission. A two-state Markov chain determines whether or not to transmit test data for each frame. The average frame activity and average burst length are defined by selecting the probabilities for transitioning between the ON/OFF states of the Markov chain, which may be driven by a second generator.

WO 01/52568 A2

METHOD AND APPARATUS FOR TESTING WIRELESS COMMUNICATION CHANNELS

BACKGROUND OF THE INVENTION

I. Field of the Invention

The present invention relates to data communication. More particularly, the present invention relates to novel and improved method and apparatus for testing wireless communication channels.

II. Description of the Related Art

Wireless communication systems such as code division multiple access (CDMA) systems, time division multiple access (TDMA) systems, and others are widely used to provide various types of communication such as voice, data, and so on. For these wireless systems, it is highly desirable to utilize the available resources (i.e., bandwidth and transmit power) as efficiently as possible. This typically entails transmitting as much data to as many users within as short a time period as supported by the conditions of the communication links.

To achieve the above goal, the communication links between a transmitting source (e.g., a base station) and the receiving devices (e.g., "connected" remote terminals) within the system may be characterized. Based on the characterized link conditions for the remote terminals, the system may be better able to select a particular set of remote terminals to serve, allocate a portion of the available resources (e.g., transmit power) to each selected remote terminal, and transmit to each remote terminal at a data rate supported by the allocated transmit power and characterized link conditions.

Conventionally, a communication link is characterized by transmitting (e.g., from a base station) a known data pattern (e.g., generated by a defined pseudo-random number generator), receiving the transmitted data pattern, comparing the received data pattern with a locally generated data pattern to determine transmission errors, and reporting the results back to the transmitting source. This "loop-back" testing is typically performed

continuously for a number of frames over the desired test interval. The test results are reflective of the performance of the communication link over that test interval.

Many newer generation wireless communication systems are capable of flexible operation. For example, data may be transmitted in bursts and over one or more traffic channels (or physical channels), the data rate may be allowed to vary from frame to frame, the processing of the data may also vary (e.g., from frame to frame and/or from channel to channel), and so on. The conventional loop-back test technique typically characterizes the communication link (e.g., one traffic channel) based on a defined set of test parameters, and may not provide an accurate assessment of the performance of the communication link when the system operates in this flexible manner.

As can be seen, techniques that can be used to characterize a communication link under various flexible operating conditions supported by a wireless communication system are highly desirable.

SUMMARY OF THE INVENTION

The present invention provides various techniques to test a wireless communication link. In one aspect, the testing of a traffic channel is performed via a test data service option (TDSO), which is a service that may be negotiated and connected using the available service configuration and negotiation procedures defined by a particular (CDMA) system and used for other services (e.g., a voice call, a data call). Values for test parameters may be proposed by an entity (e.g., a remote terminal), accepted or rejected by the other entity (e.g., a base station), and alternative values for rejected values may also be provided by the other entity. The negotiation may be performed for each traffic channel to be tested.

In another aspect, to test a traffic channel, test data is generated based on a defined data pattern or a pseudo-random number generator. Sufficient test data may be generated for a test interval (e.g., 10.24 sec) based on values from the pseudo-random number generator, and the generated test data may be stored to a (circular) buffer. Test data may thereafter be retrieved, as necessary, from a particular section of the buffer to form one or more data blocks for each

"active" frame in the test interval in which test data is to be transmitted. The particular section of the buffer from which to retrieve the test data may be identified by a particular "offset" from a current buffer pointer location, and this offset may be determined based on a number from the pseudo-random number generator. Each data block may be appropriately identified by a header to enable concurrent testing of multiple traffic channels and for testing frames having multiple data blocks per frame. In an embodiment, one pseudo-random number generator and one buffer are provided (at the transmission source and also at the receiving device) for each traffic channel, either on the forward or reverse link, to be tested.

A traffic channel may be tested using discontinuous transmission. In this case, a two-state first-order Markov chain may be used to determine whether or not to transmit test data for each frame in the test interval. By selecting the proper probabilities of transitioning between an ON state (signifying transmission of test data) and an OFF state (signifying no transmission of test data) of the Markov chain, the average frame activity and average burst length (two parameters that define a discontinuous transmission) may be defined. The Markov chain may be driven by a second pseudo-random number generator, which may be different than the one used to generate the test data.

At a receiving device, the transmitted test data is received, processed in a complementary manner, and provided to a controller. The controller further directs local generation of the test data based on a pseudo-random number generator, which is synchronized to the generator at the transmitting source. The locally generated test data is stored in a buffer and thereafter retrieved from the buffer (as necessary) and compared against the received test data. Various performance and statistical data may be collected at the remote terminal based on the results of the comparison between the received and generated test data.

The testing of the reverse link may be achieved in similar manner as that for the forward link. Multiple traffic channels on the forward and reverse links may be tested concurrently. Independent testing of the traffic channels is possible by testing each traffic channel based on a respective set of test parameter values. Thus, the forward link traffic channels and reverse link

traffic channels may be tested based on symmetric or asymmetric test parameter values. The traffic channels under test may have different frame lengths.

The invention further provides other methods and system elements that implement various aspects, embodiments, and features of the invention, as described in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

The features, nature, and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout and wherein:

FIG. 1 is a diagram of a spread spectrum communication system that supports a number of users;

FIGS. 2A and 2B are block diagrams of an embodiment of a base station and a remote terminal, respectively, capable of implementing various aspects and embodiments of the invention;

FIG. 3 is a flow diagram of a process for generating test data using a pseudo-random number generator, in accordance with a specific embodiment of the invention;

FIG. 4 is a block diagram of the buffers and pseudo-random number generators used for generating pseudo-random test data for two traffic channels;

FIG. 5 is a diagram that illustrates the reshuffling of a pseudo-random number to generate a number for the test data;

FIG. 6 is a diagram that illustrates test data transmission for a discontinuous transmission (DTX) scheme based on a deterministic frame activity;

FIG. 7 is a diagram of a two-state first-order Markov chain that may be used to model the ON/OFF states for a DTX scheme based on pseudo-random frame activity;

FIG. 8 is a flow diagram of an embodiment of a process for transitioning between the ON and OFF states of the Markov chain for a traffic channel; and

FIG. 9 is a diagram of an embodiment of a test data block.

DETAILED DESCRIPTION OF THE SPECIFIC EMBODIMENTS

FIG. 1 is a diagram of a spread spectrum communication system 100 that supports a number of users. System 100 provides communication for a number of cells, with each cell being serviced by a corresponding base station 104. Various remote terminals 106 are dispersed throughout the system. Each remote terminal 106 may communicate with one or more base stations 104 on the forward and reverse links at any particular moment, depending on whether or not the remote terminal is active and whether or not it is in soft handoff. As shown in FIG. 1, base station 104a communicates with remote terminals 106a, 106b, 106c, and 106d and base station 104b communicates with remote terminals 106d, 106e, and 106f.

A system controller 102 couples to base stations 104 and may further couple to a public switched telephone network (PSTN). System controller 102 provides coordination and control for the base stations coupled to it. System controller 102 further controls the routing of telephone calls among remote terminals 106, and between remote terminals 106 and the users coupled to PSTN (e.g., conventional telephones), via base stations 104. For a CDMA system, system controller 102 is also referred to as a base station controller (BSC).

System 100 may be designed to support one or more CDMA standards such as the "TIA/EIA-95-B Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System" (the IS-95 standard), the "TIA/EIA-98-D Recommended Minimum Standard for Dual-Mode Wideband Spread Spectrum Cellular Mobile Station" (the IS-98 standard), the "TIA/EIA/IS-2000.2-A Physical Layer Standard for cdma2000 Spread Spectrum Systems", the "TIA/EIA/IS-2000.5-A Upper Layer (Layer 3) Signaling Standard for cdma2000 Spread Spectrum Systems", the standard offered by a consortium named "3rd Generation Partnership Project" (3GPP) and embodied in a set of documents including Document Nos. 3G TS 25.211, 3G TS 25.212, 3G TS 25.213, and 3G TS 25.214 (the W-CDMA standard), the standard offered by a consortium named "3rd Generation Partnership Project 2"

(3GPP2) and embodied in a set of documents including Document Nos. C.S0002-A, C.S0005-A, C.S0010-A, C.S0011-A and C.S0026 (the cdma2000 standard), or some other standards. These standards are incorporated herein by reference.

Some newer generation CDMA systems are capable of concurrently supporting voice and data transmissions, and may further be able to transmit to a particular remote terminal via a number of forward traffic channels. For example, in the cdma2000 system, a fundamental channel may be assigned for voice and certain types of data, and one or more supplemental channels may be assigned for high-speed packet data.

FIG. 2A is a block diagram of an embodiment of base station 104, which is capable of implementing various aspects and embodiments of the invention. For simplicity, FIG. 2A shows the processing at the base station for a communication with one remote terminal. On the forward link, voice and packet data (collectively referred to herein as "traffic" data) from a transmit (TX) data source 210 and test data from a forward link (FL) test data buffer 212 are provided to a multiplexer (MUX) 214. Multiplexer 214 selects and provides the traffic data to a TX data processor 216 when operating in a normal mode, and provides the test data when operating in a test mode. TX data processor 216 receives and processes (e.g., formats, encodes, and interleaves) the received data, which is then further processed (e.g., covered, spread, and scrambled) by a modulator (MOD) 218. The modulated data is then provided to an RF TX unit 222 and conditioned (e.g., converted to one or more analog signals, amplified, filtered, and quadrature modulated) to generate a forward link signal. The forward link signal is routed through a duplexer (D) 224 and transmitted via an antenna 226 to a remote terminal.

Although not shown in FIG. 2A for simplicity, base station 104 is capable of processing and transmitting data on one or more forward traffic channels to a particular remote terminal. For a cdma2000 system, the forward traffic channels include the fundamental channel (FCH), dedicated control channel (DCCH), supplemental channel (SCH), and supplemental code channel (SCCH). The processing (e.g., encoding, interleaving, covering, and so on) for each forward traffic channel may be different from that of other forward traffic channels.

FIG. 2B is a block diagram of an embodiment of remote terminal 106. The forward link signal is received by an antenna 252, routed through a duplexer 254, and provided to an RF receiver unit 256. RF receiver unit 256 conditions (e.g., filters, amplifies, downconverts, and digitizes) the received signal and provides samples. A demodulator (DEMOD) 258 receives and processes (e.g., despreads, decovers, and pilot demodulates) the samples to provide recovered symbols. Demodulator 258 may implement a rake receiver capable of processing multiple instances of the received signal and generating combined recovered symbols. A receive (RX) data processor 260 decodes the recovered symbols, checks the received frames, and provides decoded traffic data to a RX data sink 264 and decoded test data to a controller 270. Demodulator 258 and receive data processor 260 may be operated to process multiple transmissions received via multiple forward traffic channels.

On the reverse link, a multiplexer (MUX) 284 receives results of the forward traffic channel testing from controller 270, test data for testing of the reverse link from a reverse link (RL) test data buffer 278, and traffic data from a TX data source 282. Depending on the operating mode of remote terminal 106, multiplexer 284 provides the proper combination of data and/or results to a TX data processor 286. The data and results are then processed (e.g., formatted, encoded, and interleaved) by TX data processor 286, further processed (e.g., covered, spread) by a modulator (MOD) 288, and conditioned (e.g., converted to analog signals, amplified, filtered, and quadrature modulated) by an RF TX unit 290 to generate a reverse link signal, which is then routed through duplexer 254 and transmitted via antenna 252 to one or more base stations 104.

Referring back to FIG. 2A, the reverse link signal is received by antenna 226, routed through duplexer 224, and provided to an RF receiver unit 228. The reverse link signal is conditioned (e.g., downconverted, filtered, and amplified) by RF receiver unit 228, and further processed by a demodulator 232 and an RX data processor 234 in a complementary manner to recover the transmitted data and test results. The reverse link traffic data is provided to a RX data sink 238, and the forward link test results and reverse link test data are provided to a controller 220 for evaluation.

As noted above, for efficient utilization of the available system resources, the communication link between the base station and remote terminal may be

characterized. The link characterization information may then be used to schedule data transmission, allocate transmit power, determine data rate, and so on, for the remote terminal.

The invention provides various techniques to test a wireless communication link. In an aspect, to test a forward traffic channel, test data is generated at the base station by a test data generator 240 and provided to FL test data buffer 212. The generated test data is thereafter retrieved from buffer 212 (as necessary), processed, and transmitted from the base station to the remote terminal. At the terminal, the transmitted forward link test data is received, processed in a complementary manner, and provided to controller 270. Controller 270 further directs a test data generator 280 to locally generate the test data, which is stored in a FL test data buffer 268. The locally generated test data is thereafter retrieved from buffer 268 (as necessary) and compared against the received test data. Various performance and statistical data may be collected at the remote terminal based on the results of the comparison between the received and generated test data, as described in further detail below. The testing of the reverse link may be achieved in similar manner as that for the forward link.

For clarity, various aspects of the invention are described for a specific implementation for a cdma2000 system.

<u>Channel and Frame Structure</u>

In some CDMA systems, data may be transmitted on one or more traffic channels over the forward and reverse links. (A traffic channel may be akin to a physical channel for some CDMA systems, e.g., a W-CDMA system.) For example, in a cdma2000 system, voice data is typically transmitted over a fundamental channel (FCH), traffic data is typically transmitted over a supplemental channel (SCH), and signaling may be transmitted over a dedicated control channel (DCCH). The FCH, DCCH, and SCH are different types of traffic channel. To receive a high-speed data transmission on the SCH, a remote terminal is also typically assigned a FCH or DCCH. In the cdma2000 system, each assigned traffic channel is associated with a particular radio configuration (RC) that defines the channel's transmission formats, which may

be characterized by various physical layer parameters such as the transmission rates, modulation characteristics, spreading rate, and so on.

For many CDMA systems, data is also transmitted in "frames", with each frame covering a particular time interval. For the cdma2000 system, data may be transmitted in frame lengths of 5 msec, 20 msec, 40 msec, or 80 msec on the fundamental and supplemental channels. For each frame of each connected traffic channel, one or more data blocks may be transmitted, depending on the radio configuration of the traffic channel.

In certain embodiments of the invention, the forward and reverse traffic channels are each subdivided into independent "test intervals" (which may also be referred to as "segments"). Each test interval has a duration of 10.24 seconds, which corresponds to 2048 frames for traffic channels (FCH, DCCH) with 5 msec frame length, 512 frames for traffic channels (FCH, DCCH, and SCH) with 20 msec frame length, 256 frames for traffic channels (SCH) with 40 msec frame length, and 128 frames for traffic channels (SCH) with 80 msec frame length. The first frame in the test interval is referred to as a synchronization frame. In an embodiment, the synchronization frame for each of the forward and reverse traffic channels (FCH, DCCH, SCH0, and SCH1) is selected based on (1) a 32-bit public long code mask (PLCM) assigned to the remote terminal and (2) the system frame number (SFN) of the traffic channel's frames, as described in further detail below. Thus, each traffic channel may be associated with synchronization frames that are different (time-wise) from those of other traffic channels.

In an aspect, the CDMA system is designed to support a test data service option (TDSO), which is akin to an operating mode in which the performance of the forward and/or reverse traffic channels for a remote terminal may be tested and/or verified. The initiation and negotiation of the parameters for the TDSO are described in further detail below. While operating in this mode, test data may be transmitted over the forward and/or reverse links and over one or more traffic channels on each link. This allows for independent testing of various traffic channels and further allows for independent testing of the forward and reverse links.

Test Data Generation

In accordance with an aspect of the invention, various types of test data may be used to test a traffic channel. These test data types may include defined data sequences, pseudo-random data, and others. The test data type may be selected via a parameter in the test data service option.

In one test configuration, one or more defined data sequences are used to test a traffic channel. Various schemes may be used to generate these data sequences. In one scheme, a single byte pattern is used to fill up each data block. This byte pattern may be an all ones pattern ("11111111") or some other byte pattern. If a data block includes more than a whole number of octets (e.g., 171 bits), each whole octet may be represented by the byte pattern and the remaining bits may be filled with zeros ("0"). The use of a defined data sequence may simplify the test data generation at the transmission source and receiving device.

In another test configuration, pseudo-random data is used to test a traffic channel. This data may be generated using one or more pseudo-random number generators, as described in further detail below.

FIG. 3 is a flow diagram of a process for generating test data using a pseudo-random number generator, in accordance with a specific embodiment of the invention. FIG. 3 presents an overall view of the test data generation process, which is described in greater detail below. Prior to the start of each test interval for a particular traffic channel to be tested, as determined at step 312, the pseudo-random number generators used at the transmitting source and receiving device to generate the pseudo-random test data for this traffic channel are synchronized and initialized, at step 314.

The pseudo-random number generator at the transmitting source is then operated to generate a sufficient number of test data bits for N frames (where N is two or greater), at step 316. These test data bits are stored to a (circular) buffer, which is subsequently used as the data source for bits to be packed into one or more data blocks for each "active" frame period in the test interval. The receiving device similarly generates the test data bits for N frames, which are stored to a corresponding buffer at the receiving device and thereafter retrieved

as necessary to verify whether or not the transmitted test data bits are received error free.

In accordance with an aspect of the invention and as described below, the traffic channel may be tested using discontinuous transmission. In this case, for each frame in the test interval, a TDSO state for the current frame is updated, at step 318. A determination is then made whether or not test data is to be transmitted for the current frame based on the updated TDSO state, at step 320. If test data is to be transmitted, one or more blocks of test data are retrieved from a particular section of the circular buffer, at step 322. These steps are described in further detail below.

FIG. 4 is a block diagram of the buffers and pseudo-random number generators used for generating pseudo-random test data for a forward and a reverse traffic channel, in accordance with an embodiment of the invention. In this embodiment, one pseudo-random number generator is associated with each traffic channel to be tested on each of the forward and reverse links. For example, if the TDSO is configured to transmit data over the FCH in the forward and reverse links and over the SCH0 only in the forward link, then three pseudo-random number generators are used at the base station and three pseudo-random number generators are used at the remote terminal (only two generators are shown on each side in FIG. 4).

In the embodiment shown in FIG. 4, base station 104 includes pseudo-random number generators 440a and 440b used to generate pseudo-random data for a traffic channel on the forward and reverse links, respectively. The generated test data from generators 440a and 440b is provided to test data buffers 412a and 412b, respectively. Similarly, remote terminal 106 includes pseudo-random number generators 480a and 480b used to generate pseudo-random data for the traffic channel on the forward and reverse links, respectively, which is provided to test data buffers 482a and 482b, respectively. Additional pseudo-random number generators are used for additional traffic channels to be tested. In an embodiment, pseudo-random number generators 440a, 440b, 480a, and 480b are initialized and synchronized at each synchronization frame (i.e., once every test interval), as described in further detail below.

In an embodiment, each pseudo-random number generator exhibits the following linear congruential relationship:

$$x_x = (a \cdot x_{n-1}) \bmod m \ . \qquad \text{Eq (1)}$$

In an embodiment, $a = 7^5 = 16807$, $m = 2^{31}\text{-}1 = 2{,}147{,}483{,}647$, and x_{n-1} and x_n are successive outputs of the pseudo-random number generator and are 31-bit integers. Other values may also be used for a and m.

In an embodiment, each pseudo-random number generator is initialized prior to each synchronization frame on the traffic channel associate with the generator. The initialization may be achieved as follows:

```
{
  a = 16807
  m = 2147483647
  PRNGx = seed value                  // seed the generator
  PRNGx = PRNGx XOR TOGGLE            // toggle some of the bits
  PRNGx = PRNGx AND 0x7FFFFFFF        // zero out the MSB
  PRNGx = (a•PRNGx) mod m             // iterate the generator
  PRNGx = (a•PRNGx) mod m             //    four times
  PRNGx = (a•PRNGx) mod m
  PRNGx = (a•PRNGx) mod m
}
```

In the above pseudo-code, PRNGx denotes the content of the xth pseudo-random number generator. The seed for the pseudo-random number generator may be selected as the system time, in frames, of the synchronization frame (e.g., the system frame number of the synchronization frame may be used as the seed for the pseudo-random generator). TOGGLE is a value used to toggle some of the bits of the seed, and may be selected as 0x2AAAAAAA for a generator used for the forward link and 0x55555555 for a generator used for the reverse link. As used herein, the notation "0x..." denotes a hexadecimal number.

Once initiated, the pseudo-random number generator is iterated a number of times to generate the pseudo-random test data to be used for the upcoming test interval. The number of test data bits to be generated is

dependent on various factors such as (1) the traffic channel type (i.e., FCH, DCCH, or SCH) (2) the connected radio configuration of the remote terminal, (3) the maximum number of bits to be passed by a multiplex sublayer to the physical layer for each frame period, (4) the size of the available buffer, and (5) possibly other factors. The multiplex sublayer is a protocol layer between a physical layer and a higher layer, and which multiplexes traffic data, test data, signaling, and other types of data received from the TDSO to the assigned traffic channel(s).

In an embodiment, test data bits are generated for *N* frames at the maximum bit rate possible for the connected radio configuration, as described in further detail below. A default value of two, for example, may be set for *N*, unless another value for *N* is negotiated between the base station and remote terminal. A larger value for *N* may provide test data having better randomness properties but requires a larger-sized buffer.

After initialization, the pseudo-random number generator is used to generate test data bits for *N* frames. During the test data generation, whenever a pseudo-random number is needed, the current value of the variable PRNGx is retrieved and used, and the variable PRNGx is then updated (i.e., iterated) once as shown in equation (1). In an embodiment, only the most significant 24 bits of the 31-bit number for PRNGx are used because of better randomness properties and ease of usage, and the least significant 7 bits are discarded. Thus, each iteration of the pseudo-random number generator provides a 24-bit pseudo-random number, $y_n(k)$, used to provide three bytes of test data. $P(n)$ iterations are performed to generate the required test data for *N* frames.

FIG. 5 is a diagram that illustrates a reshuffling of each pseudo-random number to generate 24 bits of test data. Using the 31-bit number from the pseudo-random number generator to generate test data is inefficient, from an implementation point of view, because the number is not octet aligned. It is easier to build a frame with a number that is octet aligned. The least significant bits of the 31-bit number are "less random" than the most significant bits, and are thus shuffled to the right. In an embodiment, each 24-bit pseudo-random number $y_n(k)$ from the pseudo-random number generator, where $1 \leq k \leq P(n)$, is reshuffled and stored in "little-endian" order. The reshuffling is achieved by

swapping the least significant byte in the 24-bit number $y_n(k)$ with the most significant byte to generate the reshuffled number $y_n^{LE}(k)$.

To generate test data for a new test interval for a particular rate $R(n)$, the TDSO generates $P(n)$ pseudo-random numbers corresponding to an actual buffer size $B(n)$, where $B(n) \geq N \bullet R(n)$. As an example, to generate 344 test data bits, the pseudo-random number generator is iterated 15 times (15•24=360, which is the first integer number of iterations that yield at least 344 bits). The buffer is then filled with the following number sequence:

$$y_n^{LE}(1),\ y_n^{LE}(2),\ y_n^{LE}(3),\ \ldots,\ y_n^{LE}(15).$$

The buffer is filled with test data at the start of each test interval and prior to the synchronization frame. Thereafter, for each "active" frame in the test interval in which test data is to be transmitted, test data bits may be retrieved from the buffer to generate one or more data blocks for the frame. For a particular traffic channel, the bits from the buffer are packed serially into one or more data blocks (e.g., corresponding to the available MUX PDU (Protocol Data Unit), as determined by the connected multiplex option, where each MUX PDU represents encapsulated data communicated between peer layers at the base station and remote terminal).

In an embodiment, the test data buffer is operated as a circular buffer and test data for each frame is retrieved from a particular section of the circular buffer (i.e., starting from a particular location in the circular buffer). Initially, after filling the circular buffer (e.g., with at least two frames of test data), a buffer pointer is set to the first location in the buffer (e.g., address zero). In an embodiment, at the start of each frame, the pseudo-random number generator is iterated once and a 24-bit number is obtained as described above. The least significant 6 bits of this 24-bit number, O_n, is then used to determine an offset for the buffer pointer. The buffer pointer is advanced from its current location by $[O_n \bmod B(n)]$ byte positions to the new starting location for the current frame. Bytes of test data are then retrieved from the circular buffer, starting from this starting location, to fill whole octets in a data block. For example, if a data block includes 171 bits, then 21 bytes (i.e., 168 bits) of test data are retrieved from the circular buffer and the remaining three bits in the data block are filled with zeros ("0").

For the next frame, the pseudo-random number generator is iterated once more, the least significant 6 bits of the 24-bit number, O_{n+1}, from the generator is used to determine the buffer pointer offset for this frame. The buffer pointer is advanced by $[O_{n+1} \bmod B(n)]$ byte positions from the current location (which is one byte position over from the last test data byte retrieved for the prior frame). This process for generating data blocks is repeated for each active frame in the test interval in which test data is to be transmitted. An example of the test data generation is provided below.

Frame and Buffer Sizes

As noted above, the pseudo-random number generator for a particular traffic channel and (forward or reverse) link to be tested is iterated a number of times (i.e., as often as necessary) to generate the test data to be used for a test interval. The number of test data bits to be generated for each test interval is based on the channel type and radio configuration. Table 1 lists the maximum number of bits for each (5 msec, 20 msec, 40 msec, or 80 msec) frame and the buffer size for the FCH and DCCH for various radio configurations defined by the cdma2000 standard.

Table 1

Reverse Radio Configuration (RC)	Forward Radio Configuration (RC)	Maximum bits/frame	Buffer Size for Two Frames (bits)	Buffer Size for *N* Frames (bits)
1, 3, 5	1, 3, 4, 6, or 7	172	2 x 172 = 344	*N* x 172
2, 4, 6	2, 5, 8, or 9	267	2 x 267 = 534	*N* x 267

Table 2 lists the maximum number of bits per frame and the buffer size for a forward supplemental channel (F-SCH0 or F-SCH1) for various radio configurations defined by the cdma2000 standard.

Table 2

Radio Configuration (RC)	Maximum bits/frame	Buffer Size for Two Frames (bits)	Buffer Size for *N* Frames (bits)
3	3,048	2 x 3,048 = 6,096	*N* x 3,048
4	6,120	2 x 6,120 = 12,240	*N* x 6,120
5	4,584	2 x 4,584 = 9,168	*N* x 4,584
6	6,120	2 x 6,120 = 12,240	*N* x 6,120
7	12,264	2 x 12,264 = 24,528	*N* x 12,264
8	9,168	2 x 9,168 = 18,384	*N* x 9,168
9	20,172	2 x 20,172 = 41,424	*N* x 20,172

Table 3 lists the maximum number of bits per frame and the buffer size for a reverse supplemental channel (R-SCH0 or R-SCH1) for various radio configurations defined by the cdma2000 standard.

Table 3

Radio Configuration (RC)	Maximum bits/frame	Buffer Size for Two Frames (bits)	Buffer Size for *N* Frames (bits)
3	6,120	2 x 6,120 = 12,240	*N* x 6,120
4	4,584	2 x 4,584 = 9,168	*N* x 4,584
5	12,264	2 x 12,264 = 24,528	*N* x 12,264
6	20,172	2 x 20,172 = 41,424	*N* x 20,172

Discontinuous Transmission Testing

In accordance with an aspect of the invention, the testing of a traffic channel may be performed in a manner to model discontinuous transmission (DTX) supported by some newer generation CDMA systems (e.g., the cdma2000 and W-CDMA systems). This DTX testing may be achieved by transmitting test data on the traffic channel in accordance with a particular ON/OFF frame activity. For each frame period (e.g., each 20 msec, 40 msec, or 80 msec) for the traffic channel, the TDSO may choose to provide to the

multiplex sublayer either one or more data blocks corresponding to a full-rate frame on that channel or one or more blank data blocks. Various DTX schemes may be used to provide data to the multiplex sublayer to achieve a particular desired frame activity. Some of these DTX schemes are described in further detail below.

In a first DTX scheme, test data is provided based on a deterministic frame activity. For this DTX scheme, test data is transmitted on the traffic channel for a particular ON duration, followed by blank data transmission for a particular OFF duration, followed by test data transmission for another ON duration, and so on. The ON and OFF durations may be selectable or negotiated between the base station and remote terminal. Also, the ON/OFF cycles may be periodic or non-periodic.

FIG. 6 is a diagram that illustrates test data transmission for an embodiment of the first DTX scheme. As shown in FIG. 6, the TDSO sends to the multiplex sublayer test data blocks for a traffic channel for a particular ON duration, and then sends blank data blocks for a particular OFF duration. The ON/OFF cycle may be designated to start at the beginning of a synchronization frame on the traffic channel being tested. The ON and OFF durations may be selected such that (1) each test interval includes one ON/OFF cycle, (2) a test interval includes multiple ON/OFF cycles, or (3) an ON/OFF cycle spans multiple test intervals.

In an embodiment, the ON duration for transmitting test data and the OFF duration for transmitting blank data may be specified by two parameters (e.g., TX_ON_PERIOD and TX_OFF_PERIOD) in a message (e.g., a *Service Option Control Message* in the cdma2000 system) sent or received by the transmitting source.

In a second DTX scheme, test data is provided in a pseudo-random manner based on a particular average frame activity and burst length. This DTX scheme may be used to achieve a particular (desired or selected) long-term average of frame activity (D) and a particular average burst length (B) for a traffic channel. The average frame activity D refers to the average number of frames in each ON duration versus the average number of frames in each ON/OFF cycle. And the average burst length B refers to the average number of frames in each ON duration.

FIG. 7 is a diagram of a two-state first-order Markov chain that may be used to model the ON/OFF states for the TDSO for the second DTX scheme. In an embodiment, one Markov chain is maintained for each traffic channel being tested. At the start of each frame, the TDSO is either in the ON state or the OFF state. The Markov chain is characterized by a probability p of transitioning from the ON state to the OFF state, and a probability q of transitioning from the OFF state to the ON state. The values of p and q may be specified by two parameters (e.g., ON_TO_OFF_PROB and OFF_TO_ON_PROB) in a message (e.g., a *Service Option Control Message*) sent by the transmitting source (e.g., the base station).

The long-term average frame activity D may be defined as:

$$D = \frac{q}{p+q} . \qquad \text{Eq (2)}$$

And the average burst length B may be defined as:

$$B = \frac{1}{p} . \qquad \text{Eq (3)}$$

For some testing, it may be desirable to select the average frame activity D and the average burst length B, and then determine the corresponding values for p and q based on the desired D and B. Combining and rearranging equations (2) and (3), the following are obtained:

$$D = \frac{Bq}{1+Bq} . \qquad \text{Eq (4)}$$

$$B = \frac{D}{(1-D)\,q} . \qquad \text{Eq (5)}$$

Equation (4) indicates that for a given value of B, D varies from 0 to $B(1+B)$ when q varies from 0 to 1, respectively. Similarly, equation (5) indicates that for a given value of D, B varies from $D/(1\text{-}D)$ to infinity when q varies from 0 to 1, respectively. For example, when B is selected as 2, D should be smaller than 2/3, which indicates that the average frame activity D cannot be set higher

than 2/3 when B is set to 2. As another example, if D is set to 7/10, then B is set greater than 7/3.

In an embodiment, a (e.g., 24-bit) pseudo-random number is used to drive the transition between the ON and OFF states for each frame period (each 5 msec, 20 msec, 40 msec, or 80 msec). In an embodiment, one pseudo-random number generator is used for all traffic channels having the same frame length. For example, one pseudo-random number generator is used for all traffic channels having 20 msec frame length. A second pseudo-random number generator is used for supplemental channels configured for 40 msec or 80 msec frame length, and this generator is updated every 40 msec or 80 msec corresponding to the channel frame length. In an embodiment, the pseudo-random number generator(s) used to drive the TDSO states are different than the ones used to generate the test data.

In an embodiment, the pseudo-random number generator(s) used to drive the transitions between TDSO states are initialized at the start of the first synchronization frame after the TDSO is initialized. Upon initialization, the Markov chain for each traffic channel is set to a particular state (e.g., OFF). The pseudo-random number generator(s) are thereafter maintained throughout the duration of the call, without reinitialization at subsequent synchronization frames. These generators may be reinitialized upon completion of a CDMA-CDMA hard handoff.

FIG. 8 is a flow diagram of an embodiment of a process for transitioning between the ON and OFF states of the Markov chain for a traffic channel. Initially, the pseudo-random number generator used to drive the TDSO states for the traffic channel is initialized, at step 812. This initialization may be achieved, for example, by obtaining a seed for the generator, XORing the seed with the value 0x2AAAAAAA, ANDing the result with the value 0x7FFFFFFF, and iterating the generator four times with the modified seed, as described in the above pseudo-code.

In an embodiment, a 24-bit pseudo-random number from the pseudo-random number generator is used to determine whether or not to transition from one state to another. Thus, 24-bit ON and OFF threshold values are computed, at step 814. These thresholds may be computed as:

$$\text{ON\_THRESHOLD} = \text{ROUND}\,(16{,}777{,}215 \bullet q), \text{ and}$$

OFF_THRESHOLD = ROUND (16,777,215 • p).

As shown in FIG. 7, the TDSO for the traffic channel transitions from the ON state to the OFF state with a probability of p, and from the OFF state to the ON state with a probability of q. Based on a pseudo-randomly generated 24-bit number, the TDSO transitions from the ON state to the OFF state if this number is less than the OFF_THRESHOLD, and from the OFF state to the ON state if this number is less than the ON_THRESHOLD. Steps 812 and 814 are typically performed once, prior to the first synchronization frame after the TDSO has been initialized.

The steps within box 820 are thereafter performed for each frame period. Initially, a 24-bit pseudo-random number is generated from the current 31-bit state of the pseudo-random number generator, at step 822. A determination is next made whether or not the current TDSO state for the traffic channel is OFF, at step 824.

If the current TDSO state is OFF, a determination is made whether the 24-bit number is greater than or equal to the ON_THRESHOLD, at step 826. If the answer is yes, the TDSO remains in the OFF state, at step 828. Otherwise, the TDSO transitions to the ON state, at step 832. In either case, the process then proceeds to step 834.

If the current TDSO state is ON (determined back at step 824), a determination is then made whether the 24-bit number is greater than or equal to the OFF_THRESHOLD, at step 830. If the answer is yes, the TDSO remains in the ON state, at step 832. Otherwise, the TDSO transitions to the OFF state, at step 828.

At step 834, the pseudo-random number generator is iterated once, as shown in equation (1), to update the state of the generator for the next frame.

Data Block Header and Format

In accordance with an aspect of the invention, each test data block is appropriately identified to enable concurrent testing of multiple traffic channels and for frames with multiple data blocks per frame. In an embodiment, the

identification is achieved via a header provided in each data block supplied to the multiplex sublayer for each frame.

FIG. 9 is a diagram of an embodiment of a test data block 900, which includes a channel ID field 912, a PDU (data block) sequence number field 914, and a test data field 916. Channel ID field 912 identifies the particular traffic channel used to send this data block. PDU sequence number field 914 identifies the sequence number of this data block within the frame (e.g., within a physical layer service data unit (SDU)). For a FCH or DCCH carrying one data block per frame, this field is set to '0'. And for an SCH capable of carrying multiple data blocks per frame, this field is set to '0' for the first data block in the SCH frame, '1' for the second data block in the SCH frame, and so on. Test data field 916 includes the (defined or pseudo-random) test data generated as described above.

Table 4 lists the fields and their lengths and definitions for an embodiment of test data block 900.

Table 4

Field	Length (bits)	Definition
Channel ID	2	Channel ID of traffic channel used to carry the data block
PDU Sequence Number	3	Sequence number of the data block within a physical layer SDU
Test Data	Variable	Test data bits

Table 5 shows a specific definition of the Channel ID field for various traffic channel types in the cdma2000 system.

Table 5

Channel ID	Traffic Channel
0	FCH
1	DCCH
2	SCH0
3	SCH1

Example of Test Data Generation

For clarity, the test data generation is now described for a specific example. In this example, the following parameters are used:

- The TDSO is configured to transmit primary traffic over the FCH.
- The base station and remote terminal are configured to support radio configuration 3, and the frame length is 172 bits.
- Multiplex option 0x01 is selected for the FCH, and one data block is passed to the multiplex sublayer for each active (20 msec) frame.
- The average frame activity D and average burst length B are based on the probabilities $p = 0.7$ and $q = 0.3$. Thus, $D = q/(p+q) = 0.3$, $B = 1/p \approx 1.4$, ON_THRESHOLD= ROUND (16,777,215 • p) = 11,744,051, and OFF_THRESHOLD = ROUND (16,777,215 • q) = 5,033,164.
- The least significant 32 bits of the remote terminal's Public Long Code Mask (PLCM) is equal to 0x9F000307.
- A first pseudo-random number generator used to determine the transitions between the ON/OFF states of the Markov chain for this traffic channel has a current value of 0x682DFF0C.

For this example, the TDSO is about to transmit frame number 0xAB89EFAD on the forward FCH (F-FCH) to the remote terminal. The frame number is XORed with the value 0x2AAAAAAA, and the least significant 9 bits of the XOR result is equal to 0x107, which is equal to the least significant 9 bits of the remote terminal's PLCM. This frame is thus the synchronization frame for the F-FCH, and the test data generation process is resynchronized.

As part of the resynchronization, a second pseudo-random number generator used to generate test data for the F-FCH is reinitialized by (1) seeding it with the frame number 0xAB89EFAD, (2) performing an XOR of the seed with the value 0x2AAAAAAA to generate the value 0x01234507, and (3) iterating the pseudo-random number generator four times, as described in the above pseudo-code.

After reinitialization, the state of the second pseudo-random number generator is 0x3B7E3E68, the most significant 24 bits of this state is 0x76FC7C, and the least significant 6 bits of this 24-bit number is 0x3C. This 6-bit number, O_n, is later used to determine the offset for the circular buffer.

number of iterations that will provide at least 344 bits included in two frames for radio configuration 3). The actual buffer size is thus $B(n) = 45$ (i.e., 360 bits = 45 bytes).

The generation of the test data proceeds as follows. Prior to each iteration, the current state of the second generator is obtained and the most significant 24 bits are used to form a 24-bit number. The following sequence of 24-bit numbers are generated by the second pseudo-random number generator:

$y_n(1)$ = 0x76FC7C	$y_n(6)$ = 0x4CA46B	$y_n(11)$ = 0xD05BFE
$y_n(2)$ = 0xBA6678	$y_n(7)$ = 0xBE783D	$y_n(12)$ = 0x478744
$y_n(3)$ = 0x9D7F54	$y_n(8)$ = 0xC7EDAF	$y_n(13)$ = 0x01A3DE
$y_n(4)$ = 0x1279A7	$y_n(9)$ = 0xC5BDB3	$y_n(14)$ = 0xAD4A7D
$y_n(5)$ = 0xF0E8EF	$y_n(10)$ = 0x29428D	$y_n(15)$ = 0xF58934

Each 24-bit number $y_n(k)$ is then stored to a circular buffer for the F-FCH in little-endian fashion, as described above. For example, the first 24-bit number 0x76FC7C is stored as 0x7CFC76, where the most and least significant bytes of the number $y_n(k)$ are swapped to generate the reshuffled number $y_n^{LE}(k)$. The circular buffer used to generate the data blocks for the F-FCH for the next 512 frames in the test interval includes the following byte sequence:

↓
→ 7C FC 76 78 66 BA 54 7F 9D A7 79 12 EF E8 F0 6B A4 4C 3D 78 BE AF ED
C7 B3 BD C5 8D 42 29 FE 5B D0 44 87 47 DE A3 01 7D 4A AD 34 89 F5 →

The first pseudo-random number generator used to determine the ON/OFF state is then updated, and a new 24-bit number having a value of 0x478744 (4,687,684) is generated. The first pseudo-random generator is updated at the end of the first iteration of the loop and after the 24-bit number is calculated, it is tested against the ON_THRESHOLD during the second iteration around the loop. Since this value is less than the ON_THRESHOLD value of 11,744,051, the TDSO transitions from the OFF state to the ON state, and a data block is provided to the multiplex sublayer for the current frame.

To generate this data block for the first frame in the test interval, the offset for the buffer pointer is computed as O_n mod $B(n)$ (i.e., 0x3C mod 45 = 60 mod 45 = 15). The buffer pointer (which is initialized to zero upon

reinitialization) is thus advanced by 15 byte positions, from 0x7C to 0x6B. The 171 bits for the data block are then formed with 21 bytes (168 bits) retrieved from the circular buffer, starting at the buffer location identified by the advanced buffer pointer. The remaining three bits in the data block are filled with zeros. The data block includes the following byte sequence:

```
6B A4 4C 3D 78 BE AF ED C7 B3 BD C5 8D 42 29 FE 5B D0 44 87 47 '000'
```

Since this frame is to be sent over the F-FCH, the first 5 bits of the octet are replaced by '00000' corresponding to the channel ID of '00' and the PDU sequence number of '000'. The final test data block is as follows:

```
03 A4 4C 3D 78 BE AF ED C7 B3 BD C5 8D 42 29 FE 5B D0 44 87 47 '000'
```

For the next TDSO frame, a new 24-bit number having a value of 107,486 is generated by the first pseudo-random number generator. Since this value is less than the ON threshold, the TDSO remains in the ON state and a new data block is generated for the multiplex sublayer.

For the second frame in the test interval, the second pseudo-random number generator is iterated, and a 24-bit number having a value of 0x02F3FD is generated. The 6-bit number O_n for the buffer offset has a value of 0x3D. The buffer offset is then computed as O_n mod $B(n)$ (i.e., 0x3D mod 15 = 61 mod 45 = 16). The buffer pointer (which was pointing one byte location over from the last retrieved byte value of 0x47 for the last data block) is thus advanced by 16 byte positions from 0xDE to 0x6F. The 171 bits for the data block are then formed with 21 bytes from the circular buffer, starting at the new buffer location. The remaining three bits in the data block are filled with zeros. The data block includes the following byte sequence:

```
7F 9D A7 79 12 EF E8 F0 6B A4 4C 3D 78 BE AF ED C7 B3 BD C5 8D '000'
```

After replacing the first 5 bits with '00000' corresponding to the data block header for the F-FCH, the data block provided to the multiplex sublayer is as follows:

```
07 9D A7 79 12 EF E8 F0 6B A4 4C 3D 78 BE AF ED C7 B3 BD C5 8D '000'
```

The buffer pointer now points to the next byte position (0x42) for the next frame.

TDSO Frame Transmission and Reception

To test a particular traffic channel, the data block(s) for each "active" frame are generated based on a defined data pattern or a pseudo-random number generator, as described above. The transmitting source and receiving device are synchronized so that the receiving device is able to properly generate the transmitted frames, such that the received frames may be compared with the locally generated frames. Each data block in each frame is appropriately identified to indicate (1) the particular traffic channel used to send the data block and (2) the data block number within the frame. The TDSO is able to compare the received and locally generated frames, count the errors, determine the bit error rate (BER), PDU or data block error rate (PER), and frame error rate (FER), and compute other measures of performance.

The testing thus includes processing performed at the transmitting source to transmit a test frame and processing performed at the receiving device to receive a test frame.

The transmit frame processing includes:

- Generating one or more data blocks for each active frame.
- Supplying the generated data block(s) to the multiplex sublayer for transmission.
- Incrementing the appropriate counters.

For a test of the FCH or DCCH that operates on 20 msec frames, the TDSO provides one data block to the multiplex sublayer for each active frame interval in which the TDSO state for the traffic channel is ON. For a test of the SCH, the TDSO provides N_B data blocks to the multiplex sublayer for each active frame interval (20 msec, 40 msec, or 80 msec), where N_B is the maximum number of data blocks in a physical layer SDU for the connected service option. Each data block may be generated as described above, and includes the header and test data.

The receive frame processing includes:

- Generating one or more data blocks for each active frame.
- Receiving data block(s) from the multiplex sublayer.
- Comparing the rates and contents of the received and generated data block(s).

- Incrementing the appropriate counters.

At the receiving device, the multiplex sublayer categorizes each received data block (e.g., as either test data or blank) and the frame. The multiplex sublayer then supplies the data block type and received test data bits, if any, to the TDSO.

Various counters may be maintained at the transmitting source and receiving device to support TDSO. For each traffic channel to be tested, a set of counters may be maintained at the transmitting source to keep track of the number of frames (of various types) and data blocks transmitted to the receiving device. At the receiving device, another set of counters may be maintained to keep track of the number of frames, data blocks, and data bits received from the transmitting source, the number of frame errors, block errors, and bit errors, and so on. These counter values may be stored in a buffer. This buffer is typically implemented separate from the data buffer, and is used to store various counters over a period of time. The counter values may thereafter be used to determine the FER, PER, and/or BER, and other statistics such as the average frame activity, average burst length, and so on. The test results and statistical information may be reported from the remote terminal to the base station via one or more messages.

<u>Test Data Service Option</u>

In accordance with an aspect of the invention, the test data service option (TDSO) is a service that may be negotiated and connected using the available service configuration and negotiation procedures defined by a particular CDMA system and used for other services (e.g., a voice call, a data call). The remote terminal may be able to propose and/or accept a service configuration having attributes that are consistent with valid attributes for that configuration. The remote terminal may also be able to indicate the preferred radio configurations for the forward and reverse links.

In an embodiment, the remote terminal is able to propose or invoke service-option-specific functions for a TDSO call by sending a message (e.g., a *Service Option Control Message* in the cdma2000 system) to the base station. This message may be sent such that an acknowledgement is requested or required from the base station. Via the message, the remote terminal may propose values for various test parameters to be used during the test period.

The base station receives the message and may accept or reject the remote terminal's proposed test parameter settings. If all the fields in the remote terminal's directive are within acceptable ranges for the base station, the base station may issue a directive that accepts the remote terminal's proposal. This directive may be sent to the remote terminal via a response message (e.g., a *Service Option Control Message*) that includes the same values, as proposed by the remote terminal, for the various fields.

Alternatively, if the remote terminal proposes a particular test setting not supported by or acceptable to the base station, the base station may issue a directive that may include alternative values (i.e., counter-proposals) to the remote terminal's proposed values. This directive may be sent to the remote terminal via a response message that includes the proposed values in the fields supported and accepted by the base station, and counter-proposed values in the fields not supported or accepted by the base station. For example, if the remote terminal requests a particular number of circular buffer frames *N* that is not supported by the base station, the base station may response with a value indicating the maximum number of frames for the buffer supported by the base station.

Thus, via messaging and negotiation, the base station is able to accept the remote terminal's proposal, or reject the proposal and provide alternative values for test parameters.

Upon receiving the response message from the base station, the remote terminal may accept the counter-proposed values or select new values that conform to the counter-proposed values. The remote terminal may then send to the base station another message proposing these new values.

Table 6 lists the valid service configuration for TDSO for a specific implementation in the cdma2000 system.

Table 6

Service Configuration Attribute	Valid Selection
Forward Multiplex Option	0x01 or 0x02
Reverse Multiplex Option	0x01 or 0x02
Forward Transmission Rates	For the FCH - Rates 1, 1/2, 1/4, and 1/8 enabled For the DCCH - Rate 1 enabled, Rates 1/2, 1/4, and 1/8 not enabled
Reverse Transmission Rates	For the FCH, Rates 1, 1/2, 1/4, and 1/8 enabled. For the DCCH, Rate 1 enabled, Rates 1/2, 1/4, and 1/8 not enabled.
Forward Traffic Type	Primary or Secondary
Reverse Traffic Type	Should be Identical to the Forward Traffic Type
Forward FCH Radio Configuration	RC 1, 2, 3, 4, 5, 6, 7, 8, or 9
Reverse FCH Radio Configuration	RC 1, 2, 3, 4, 5, or 6
Forward DCCH Radio Configuration	RC 3, 4, 5, 6, 7, 8, or 9
Reverse DCCH Radio Configuration	RC 3, 4, 5, or 6
Forward SCH Radio Configuration	RC 3, 4, 5, 6, 7, 8, or 9
Reverse SCH Radio Configuration	RC 3, 4, 5, or 6
Forward SCH Frame Size	20 ms, 40 ms, or 80 ms
Reverse SCH Frame Size	20 ms, 40 ms, or 80 ms
Forward Supplemental Channel Multiplex Option	0x921, 0x911, 0x909, 0x905, 0x821, 0x811, 0x809, 0x03 0x922, 0x912, 0x90a, 0x906, 0x822, 0x812, 0x80a, 0x04, 0xf20
Reverse Supplemental Channel Multiplex Option	0x921, 0x911, 0x909, 0x905, 0x821, 0x811, 0x809, 0x03 0x922, 0x912, 0x90a, 0x906, 0x822, 0x812, 0x80a, 0x04, 0xf20

As noted above, a number of traffic channels may be concurrently tested on each of the forward and reverse links. For each traffic channel to be tested, the test parameters for the channel may be negotiated via the signaling and negotiation described above. Thus, traffic channels of various types on the forward and reverse links may be tested independently based on their respective sets of test parameter values.

In FIGS. 2A, 2B, and 4, the elements in the base station and remote terminal may be implemented by various means. For example, the pseudo-random number generators may be implemented with hardware, software, or a combination thereof. For a hardware implementation, pseudo-random number generators, controllers, and other processing units may be implemented within one or more application specific integrated circuits (ASICs), digital signal processors (DSPs), programmable logic devices (PLDs), controllers, micro-controllers, microprocessors, other electronic units designed to perform the functions described herein, or a combination thereof.

For a software implementation, these processing units may be implemented with modules (e.g., procedures, functions, and so on) that perform the functions described herein. For example, the pseudo-random number generators may be implemented with software code stored in a memory unit and executed by a processor (e.g., controller 220 or 270).

The circular buffers for the test data for the traffic channels may be implemented with one or more buffers, which may be implemented using RAM, DRAM, Flash memory, or some other memory technology. Also, the pseudo-random number generators may be operated to generate test data for the traffic channels as the data is needed, without having to store the test data in buffers. In that case, the states of the pseudo-random number generators are appropriately maintained and updated such that the generators are able to generate the proper sequence of test data for each active frame.

Although various aspects, embodiments, and features of the test data generation and traffic channel testing of the invention have been described for the cdma2000 system, these techniques may be advantageously applied for the other wireless communication systems and other CDMA systems (e.g., the W-CDMA system).

A specific implementation of various aspects of the invention for a cdma2000 system is described in the following Exhibit A.

Equation 3 shows that given a fixed B, D varies from 0 to $B/(1+B)$, when q varies from 0 to 1. Similarly, Equation 4 shows that given a fixed value of D, B varies from $D/(1-D)$ to infinity.

For example, if B is set to 2, D has to be smaller than 2/3. As a result, the frame activity (D) can never get higher than 2/3 when B is set to 2. Similarly, if D is set to 7/10, B has to be greater than 7/3.

The corresponding valid values of p and q can be calculated from Equation 1 and Equation 2 given a valid pair of D and B.

The foregoing description of the preferred embodiments is provided to enable any person skilled in the art to make or use the present invention. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without the use of the inventive faculty. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

WHAT IS CLAIMED IS:

CLAIMS

1. A method for generating test data for testing a particular channel in a wireless communication system, comprising:

generating a sequence of data bits based on a pseudo-random number generator; and

forming a plurality of data blocks for transmission over a plurality of time intervals on the particular channel, wherein each data block includes at least a portion of the generated sequence of data bits.

2. The method of claim 1, wherein each time interval corresponds to a frame on the particular traffic channel, and wherein the sequence of data bits includes at least N times a maximum number of bits expected to be transmitted for one fame on the particular channel, where N is two or greater.

3. The method of claim 1, further comprising:

storing the generated sequence of data bits to a buffer.

4. The method of claim 3, wherein the buffer is operated as a circular buffer, the method further comprising:

retrieving data bits for each data block from a particular section of the circular buffer.

5. The method of claim 4, wherein a starting location in the circular buffer from which to retrieve data bits for a particular data block is determined based in part on a value obtained from the pseudo-random number generator.

6. The method of claim 5, further comprising:

formatting the value obtained from the pseudo-random number generator; and

advancing a pointer for the circular buffer by a number of positions determined based on the formatted number.

7. The method of claim 6, wherein a 31-bit value is obtained from the pseudo-random number generator, and wherein the formatting includes

generating a 24-bit number with 24 most significant bits of the 31-bit value, and

generating the formatted number with six least significant bits of the 24-bit number.

8. The method of claim 1, wherein the generating the sequence of data bits includes

obtaining a value corresponding to a current state of the pseudo-random number generator,

forming a set of data bits based on the obtained value, and

updating the pseudo-random number generator.

9. The method of claim 8, wherein the generating the sequence of data bits further includes

repeating the obtaining, forming, and updating a plurality of times, and

concatenating a plurality of sets of data bits formed based on a plurality of values obtained from the pseudo-random number generator to generate the sequence of data bits.

10. The method of claim 8, wherein the forming includes

extracting a most significant portion of the obtained value, and

rearranging bytes in the extracted most significant portion to form the set of data bits.

11. The method of claim 10, wherein a 31-bit value is obtained from the pseudo-random number generator, a 24-bit value is extracted from the most significant portion of the obtained value, and the bytes of the 24-bit value are rearranged in little-endian order.

12. The method of claim 1, further comprising:

reinitializing the pseudo-random number generator at each synchronization time corresponding to a start of a new test interval.

13. The method of claim 12, wherein each test interval has a duration of 10.24 seconds.

14. The method of claim 12, wherein the synchronization time is determined based in part on a system frame number for a frame on the particular traffic channel.

15. The method of claim 14, wherein the synchronization time is further determined based on a public long code mask (PLCM) assigned to a remote terminal designated to receive the data blocks.

16. The method of claim 1, wherein a plurality of channels are concurrently tested, and wherein a plurality of pseudo-random number generators are used to generate test data for testing the plurality of channels.

17. The method of claim 16, wherein one pseudo-random number generator is used to generate test data for each channel.

18. The method of claim 17, wherein the test data generated for each channel is stored to a respective buffer.

19. A method for generating test data for testing a particular channel in a wireless communication system, comprising:

selecting a particular one of a plurality of available test data types;

generating a sequence of data bits of the selected test data type; and

forming a plurality of data blocks for transmission over a plurality of time intervals on the particular channel, wherein each data block includes at least a portion of the generated sequence of data bits.

20. The method of claim 19, wherein the available test data types include test data generated based on a defined data pattern and test data pseudo-randomly generated.

21. The method of claim 20, wherein the sequence of data bits generated based on the defined data pattern includes a plurality of bytes of a particular value.

22. The method of claim 21, wherein the defined data pattern is a sequence of a particular number of ones.

23. A method for testing a particular channel in a wireless communication system, comprising:

determining a transmission state of a current frame for the particular channel, wherein transmission on the particular channel occurs over frames, and wherein each frame corresponds to a particular time interval;

generating one or more blocks of test data for the current frame if the determined transmission state indicates that test data is to be transmitted; and

transmitting the one or more generated blocks of test data on the particular channel.

24. The method of claim 23, further comprising:

maintaining a two-state Markov chain to represent the transmission state for the particular channel.

25. The method of claim 24, wherein the two-state Markov chain includes an ON state signifying transmission of test data on the particular channel and an OFF state signifying no transmission of test data on the particular channel.

26. The method of claim 25, further comprising:

maintaining a pseudo-random number generator to determine transitions between the ON and OFF states of the Markov chain.

27. The method of claim 26, further comprising:

initializing the pseudo-random number generator prior to start of testing the particular channel.

28. The method of claim 26, further comprising:

obtaining a value based on a current state of the pseudo-random number generator; and

transitioning from the ON state to the OFF state if a current state of the Markov chain is the ON state and the obtained value is below a first threshold value.

29. The method of claim 28, further comprising:

transitioning from the OFF state to the ON state if the current state of the Markov chain is the OFF state and the obtained value is below a second threshold value.

30. The method of claim 29, wherein the first and second thresholds values are configurable test parameters.

31. The method of claim 25, wherein transition between the ON state and the OFF state is based on a first probability and transition between the OFF state and the ON state is based on a second probability.

32. The method of claim 31, wherein the first and second probabilities are selected to achieve a particular average frame activity on the particular channel indicative of an average duty cycle for transmissions on the channel.

33. The method of claim 32, wherein the average frame activity is a selectable test parameter.

34. The method of claim 31, wherein the first and second probabilities are selected to achieve a particular average burst length on the particular channel indicative of an average duration for transmissions on the channel.

35. The method of claim 23, wherein transmission of test data occurs on the particular channel for a particular ON duration followed by no transmission of test data for a particular OFF duration.

36. The method of claim 35, wherein the ON and OFF durations are configurable test parameters.

37. The method of claim 26, wherein a plurality of channels are concurrently tested, and wherein a two-state Markov chain is maintained for each channel being tested.

38. The method of claim 37, wherein one pseudo-random number generator is maintained to determine transitions between Markov states for each set of one or more channels having a frame interval that is different from frame intervals of other channels being tested.

39. The method of claim 37, wherein a first pseudo-random number generator is maintained to determine transitions between Markov states for a first set of one or more channels having a first frame interval, and wherein a second pseudo-random number generator is maintained to determine transitions between Markov states for a second set of one or more channels having a second frame interval.

40. The method of claim 39, wherein the first frame interval is 20 msec and the second frame interval is 40 msec or 80 msec.

41. A method for testing a plurality of channels in a wireless communication system, comprising:

defining values for a set of test parameters for each of the plurality of channels to be tested; and

testing each of the plurality of channels in accordance with respective values defined for the set of test parameters.

42. The method of claim 41, wherein the plurality of channels have two or more different frame lengths.

43. The method of claim 41, wherein the plurality of channels have frame lengths selected from the group consisting of 5 msec, 20 msec, 40 msec, and 80 msec.

44. The method of claim 41, wherein the plurality of channels include at least one forward traffic channel and at least one reverse traffic channel.

45. The method of claim 41, further comprising:

generating data blocks for transmission over a plurality of frames on the plurality of channels, wherein each data block includes a header that identifies the particular channel on which the data block is transmitted.

46. The method of claim 41, wherein each traffic channel to be tested is associate with a respective sequence of test data bits.

47. The method of claim 41, wherein each traffic channel to be tested is associate with a respective average frame activity.

48. The method of claim 41, wherein each traffic channel to be tested is associate with a respective average burst length.

49. The method of claim 41, further comprising:

maintaining a two-state Markov chain to represent a transmission state for each of the plurality of channels, wherein the two-state Markov chain for

each channel includes an ON state signifying transmission of test data on the channel and an OFF state signifying no transmission of test data on the channel.

50. The method of claim 49, further comprising:

maintaining one or more pseudo-random number generators to determine transitions between the ON and OFF states of Markov chains for the plurality of channels.

51. The method of claim 50, wherein one pseudo-random number generator is maintained for each set of one or more channels having same frame length.

52. The method of claim 51, wherein a first pseudo-random number generator is maintained for one or more channels having frame length of 20 msec and a second pseudo-random number generator is maintained for one or more channels having frame length of 40 msec or 80 msec.

53. A method for testing a particular channel in a wireless communication system, comprising:

sending from a first entity to a second entity a first message having included therein one or more proposed values for one or more parameters for testing the particular channel; and

receiving from the second entity a response message rejecting or accepting the one or more proposed values sent in the first message.

54. The method of claim 53, wherein the response message includes one or more alternative values for one or more parameters rejected by the second entity.

55. The method of claim 53, further comprising:

sending to the second entity a second message having included therein one or more values for one or more parameters rejected by the second entity.

56. The method of claim 53, wherein the first entity is a remote terminal and the second entity is a base station in the communication system.

57. A transmitting entity in a wireless communication system, comprising:

at least one pseudo-random number generator, each generator configured to generate pseudo-random numbers used to generate a sequence of data bits; and

at least one buffer operatively coupled to the at least one generator, each buffer configured to store a respective generated sequence of data bits, and

wherein a plurality of data blocks are formed for transmission over a plurality of time intervals on a particular channel, and wherein each data block includes at least a portion of a particular sequence of data bits from a particular buffer.

58. The transmitting entity claim 57, further comprising:

a controller configured to select one of a plurality of available test data types, wherein the available test data types include test data generated based on a defined data pattern and test data pseudo-randomly generated.

59. The transmitting entity claim 58, wherein the controller is further configured to determine a transmission state of a current frame for the particular channel, and wherein the transmission state is either an ON state signifying transmission of test data on the particular channel in the current frame or an OFF state signifying no transmission of test data on the particular channel in the current frame.

60. The transmitting entity claim 57, wherein a plurality of channels are concurrently tested, and wherein one pseudo-random number generator and one buffer are associate with each channel to be tested.

61. In a wireless communication system in which a plurality of frames are transmitted, a method for attaining a long-term average value on a duty cycle using a two-state Markov chain, the method comprising:

driving on/off transitions of a test data service option (TDSO) process with a first pseudo-random number generator during a frame period if the frame period is a first length in time; and

driving the on/off transitions with a second pseudo-random number generator during the frame period if the frame period is either a second length in time or a third length in time.

62. The method of claim 61, wherein the first and second pseudo-random number generators provide 24-bits pseudo-random numbers.

63. The method of claim 61, wherein the first length in time is 20 msec.

64. The method of claim 61, wherein the second length in time is 40 msec and the third length in time is 80 msec.

65. The method of claim 61, wherein if the frame period is equal to either the second length in time or the third length in time, the frame is a supplemental channel.

66. The method of claim 61, wherein the long-term average value is configurable.

67. A method of exchanging test parameter values between a remote terminal and a base station in a wireless communication system, the method comprising:

sending proposed test parameter values from the remote terminal to the base station; and

receiving a service option control message from the base station rejecting or negatively acknowledging the proposed test parameter values.

68. A method of constructing a circular buffer storing a plurality of maximum-rate frames transmitted on a particular channel in a wireless communication system, the method comprising:

constructing data for the circular buffer from iterations of a pseudo-random number generator a plurality of times for each test interval; and

using a set of bits from a number generated by the pseudo-random number generator to indicate a byte offset to determine a starting position in the circular buffer from which to build one or more data blocks for a particular frame period.

69. The method of claim 68, wherein the pseudo-random number generator is a 31-bit pseudo-random number generator.

70. The method of claim 68, wherein the set of bits is obtained by

extracting 24 most significant bits of number generated by the pseudo-random number generator, and

extracting six least significant bits of the 24 most significant bits.

71. The method of claim 68, wherein the test interval is defined to coincide with a synchronization frame of the channel.

72. The method of claim 71, wherein the test interval has a duration of 10.24 seconds.

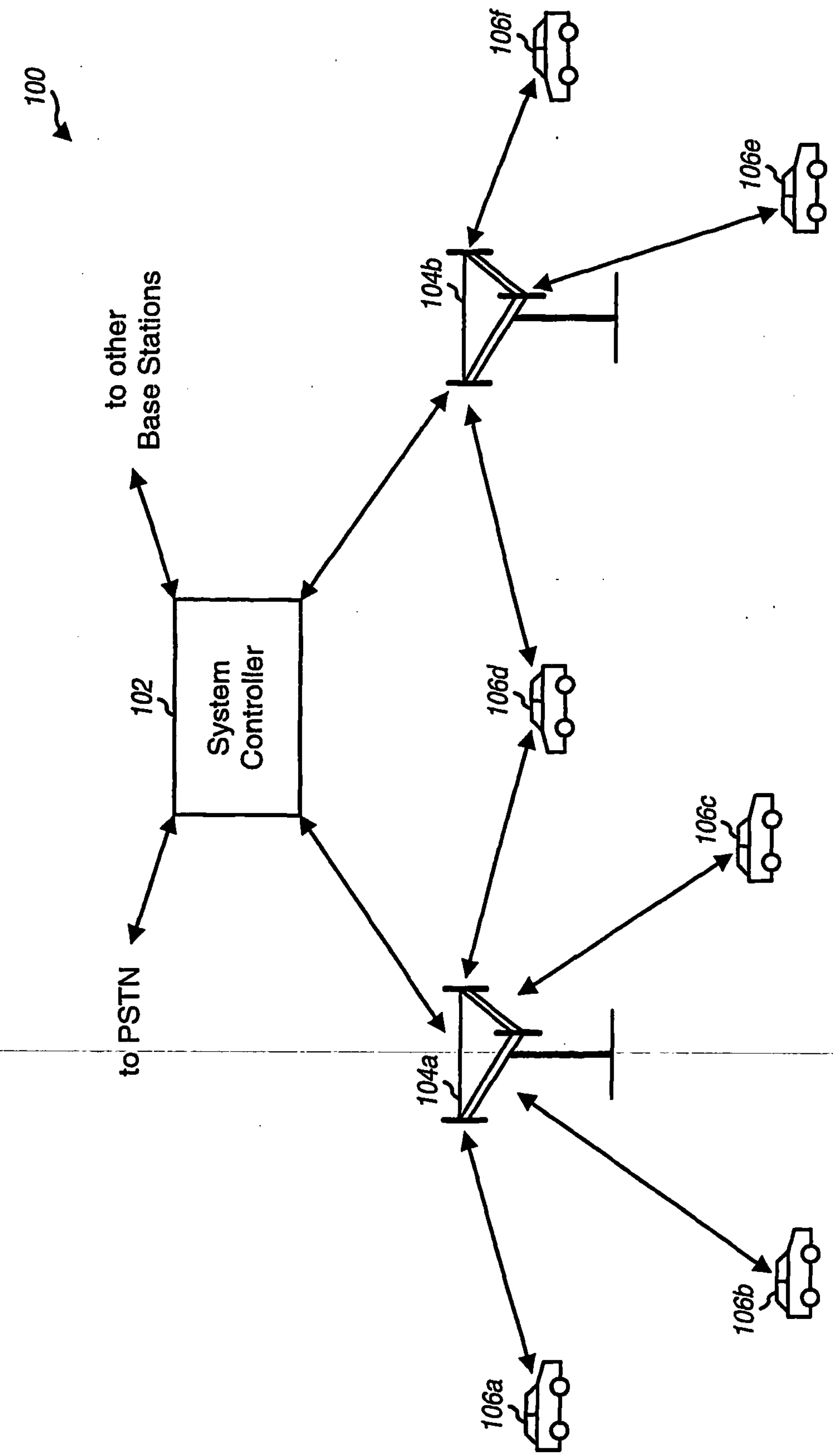
100
to other
Base Stations
to PSTN
102
System
Controller
104a
104b
106a
106b
106c
106d
106e
106f

FIG. 1

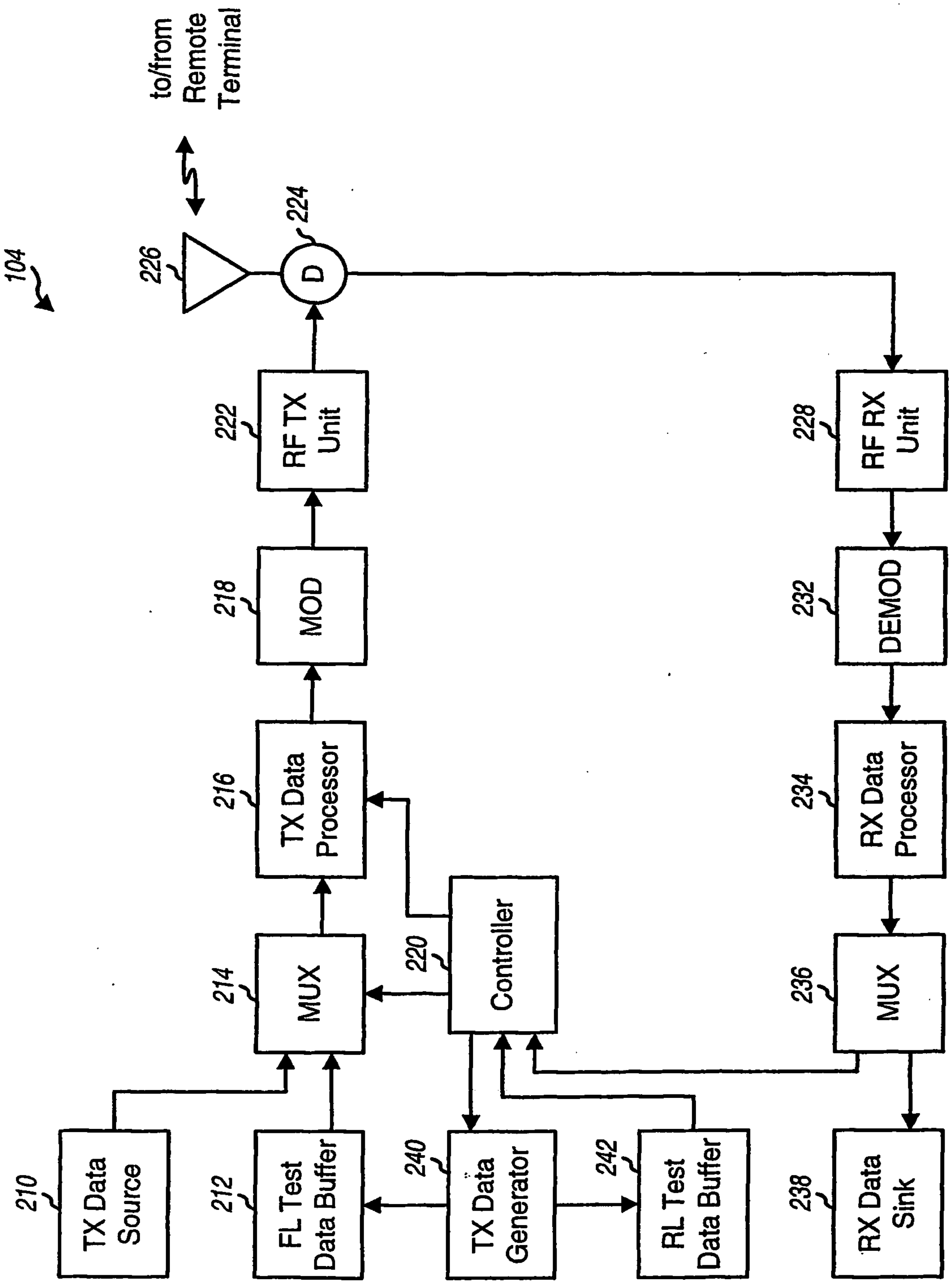
104
to/from Remote Terminal
226
224
D
222
RF TX Unit
218
MOD
216
TX Data Processor
214
MUX
220
Controller
210
TX Data Source
212
FL Test Data Buffer
240
TX Data Generator
242
RL Test Data Buffer
228
RF RX Unit
232
DEMOD
234
RX Data Processor
236
MUX
238
RX Data Sink

FIG. 2A

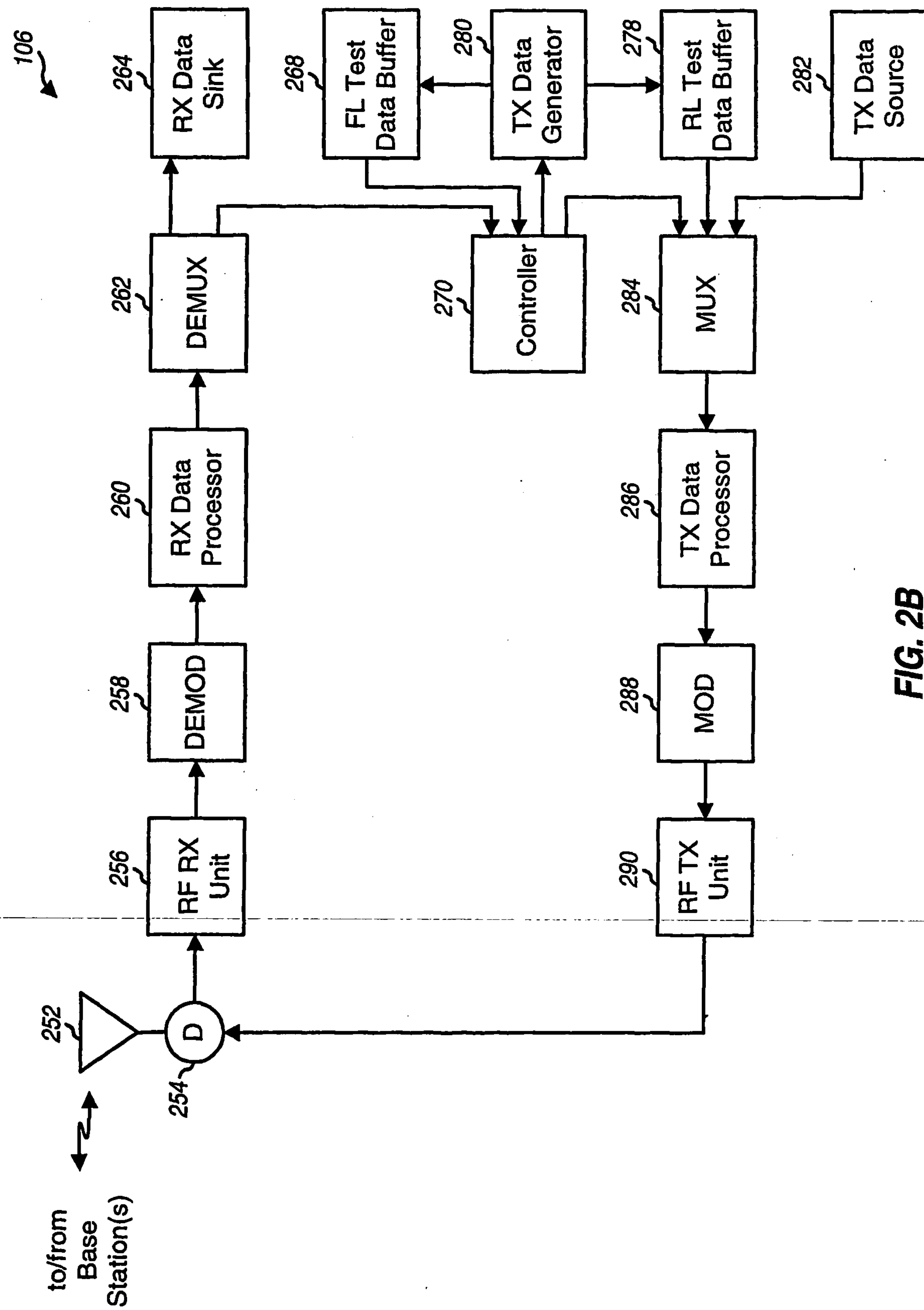
106
264
RX Data Sink
268
FL Test Data Buffer
280
TX Data Generator
278
RL Test Data Buffer
282
TX Data Source
262
DEMUX
270
Controller
284
MUX
260
RX Data Processor
286
TX Data Processor
258
DEMOD
288
MOD
256
RF RX Unit
290
RF TX Unit
252
D
254
to/from Base Station(s)

FIG. 2B

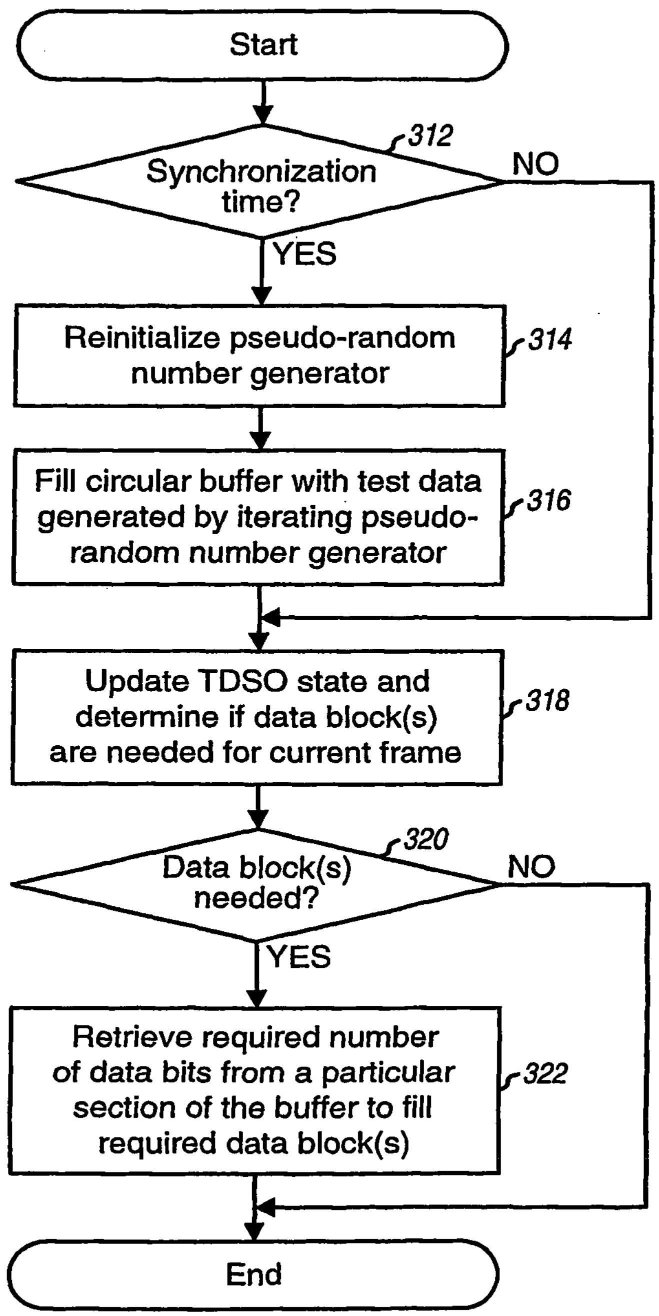
Start
312
Synchronization time?
NO
YES
Reinitialize pseudo-random number generator
314
Fill circular buffer with test data generated by iterating pseudo-random number generator
316
Update TDSO state and determine if data block(s) are needed for current frame
318
320
Data block(s) needed?
NO
YES
Retrieve required number of data bits from a particular section of the buffer to fill required data block(s)
322
End

FIG. 3

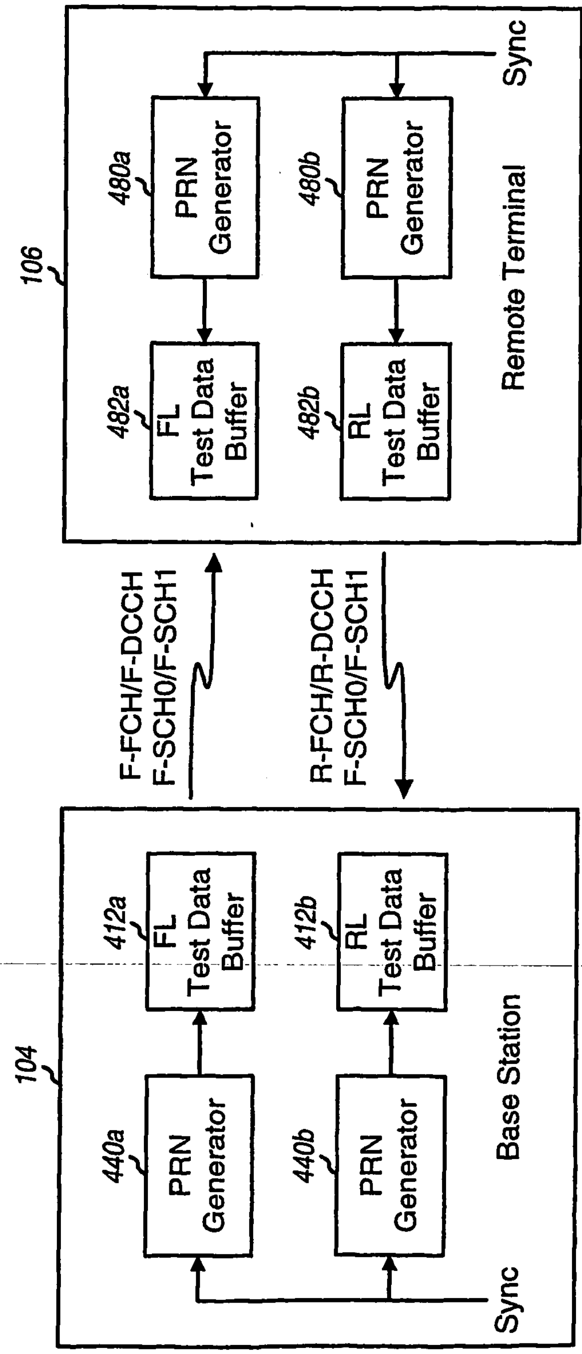
106
480a
PRN Generator
480b
PRN Generator
482a
FL Test Data Buffer
482b
RL Test Data Buffer
Sync
Remote Terminal
F-FCH/F-DCCH
F-SCH0/F-SCH1
R-FCH/R-DCCH
F-SCH0/F-SCH1
104
412a
FL Test Data Buffer
412b
RL Test Data Buffer
440a
PRN Generator
440b
PRN Generator
Base Station
Sync

FIG. 4

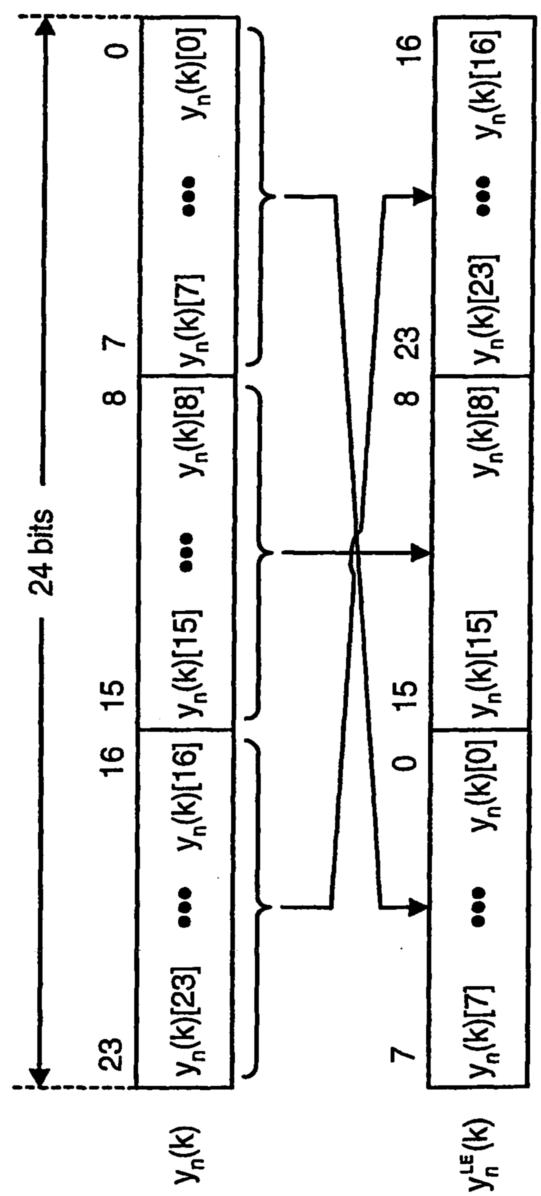
24 bits
$y_n(k)$
$y_n^{LE}(k)$
23
16 15
8 7
0
$y_n(k)[23]$
$y_n(k)[16]$
$y_n(k)[15]$
$y_n(k)[8]$
$y_n(k)[7]$
$y_n(k)[0]$
7
0 15
8 23
16

FIG. 5

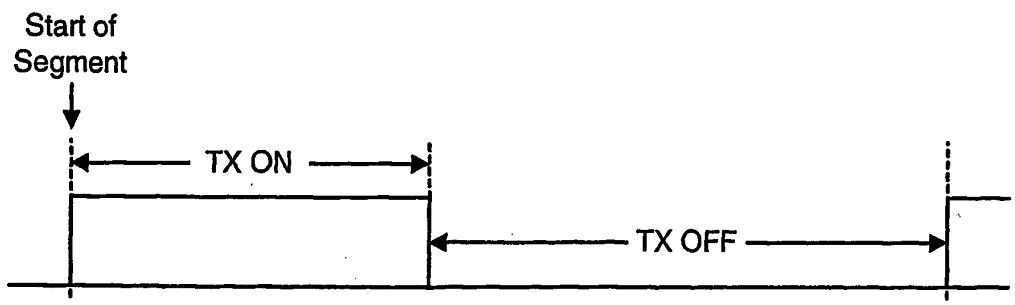

FIG. 6

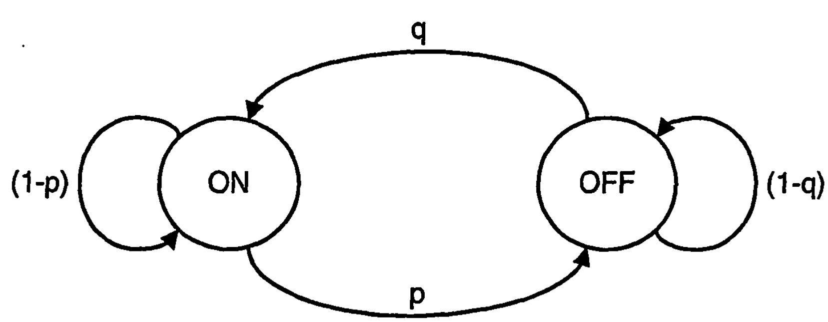

FIG. 7

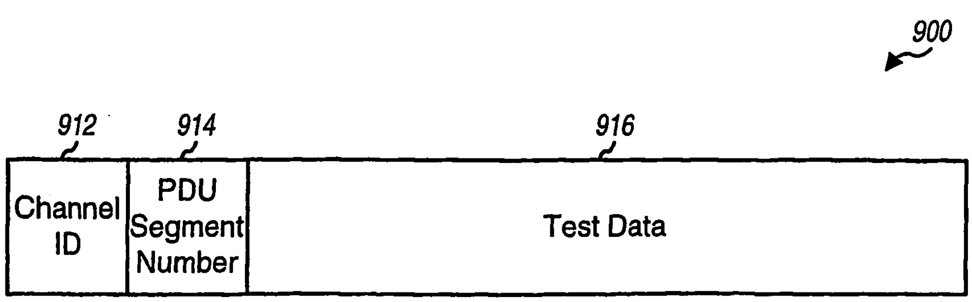

FIG. 9

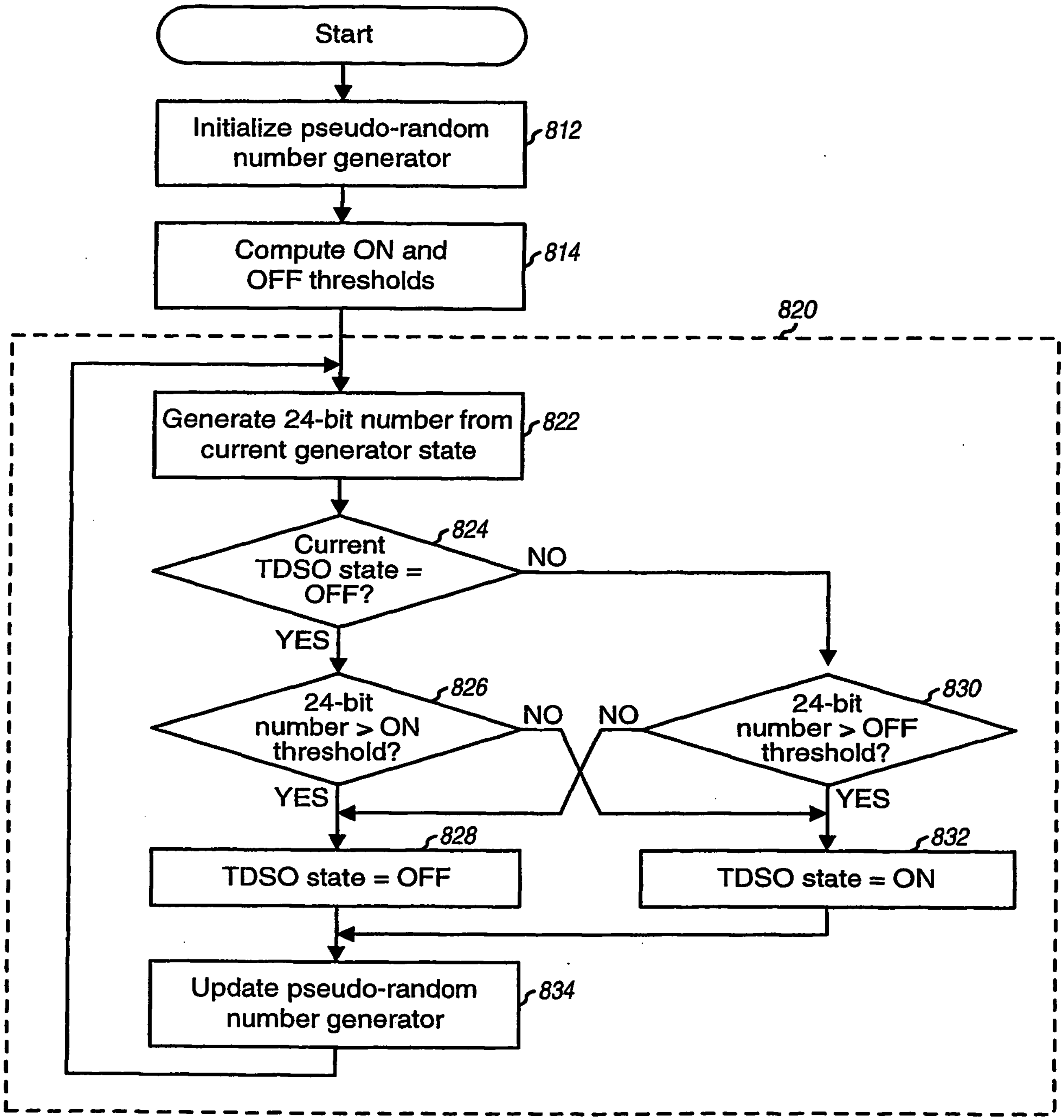
Start
Initialize pseudo-random number generator
812
Compute ON and OFF thresholds
814
820
Generate 24-bit number from current generator state
822
Current TDSO state = OFF?
824
NO
YES
24-bit number > ON threshold?
826
NO
NO
24-bit number > OFF threshold?
830
YES
YES
TDSO state = OFF
828
TDSO state = ON
832
Update pseudo-random number generator
834

FIG. 8